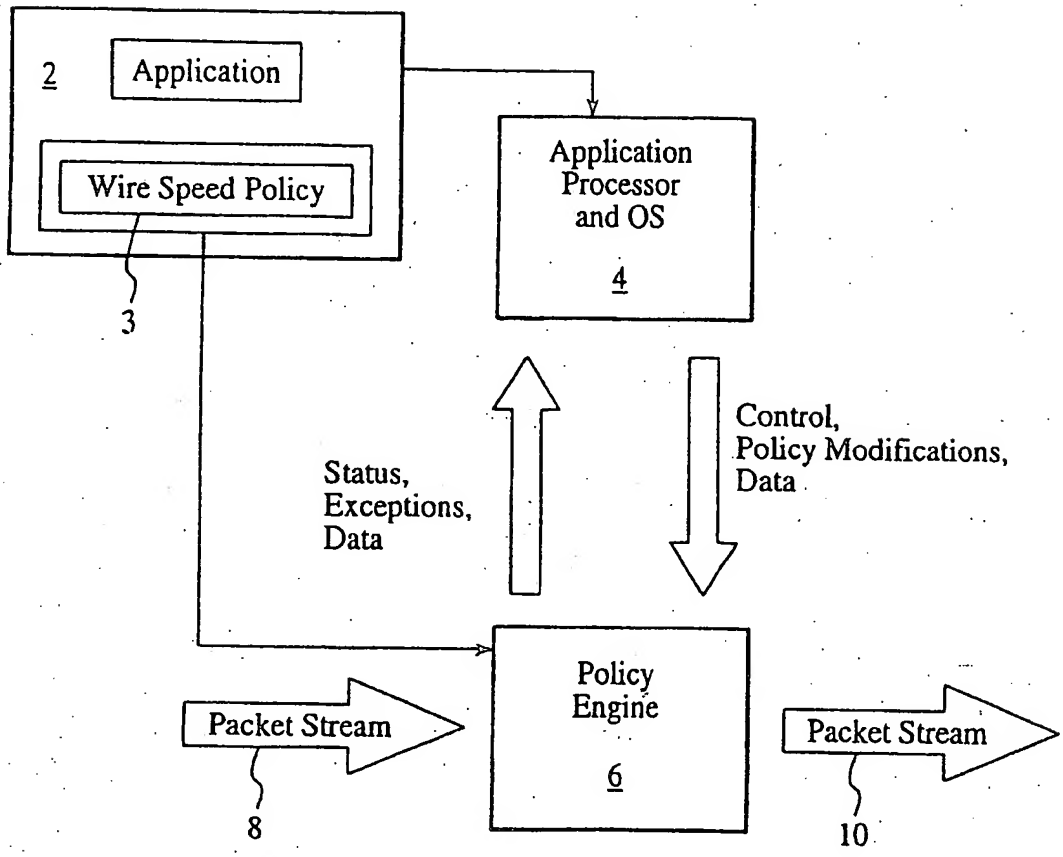

2
Application
Wire Speed Policy
3
Application Processor and OS
4
Status, Exceptions, Data
Control, Policy Modifications, Data
Policy Engine
6
Packet Stream
8
Packet Stream
10

FIG. 1

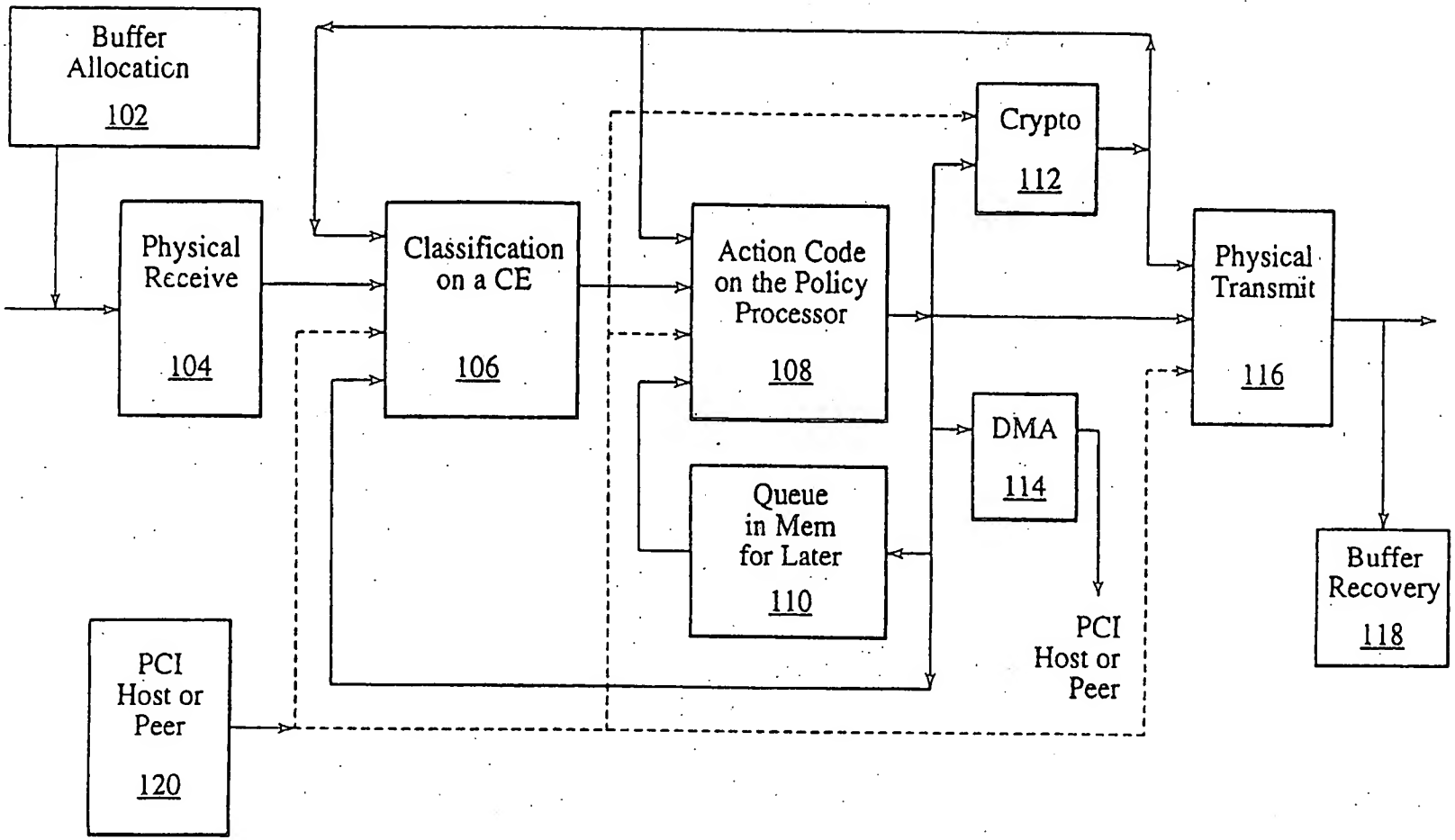

Buffer Allocation
102
Physical Receive
104
Classification on a CE
106
Action Code on the Policy Processor
108
Queue in Mem for Later
110
Crypto
112
DMA
114
Physical Transmit
116
Buffer Recovery
118
PCI Host or Peer
120
PCI Host or Peer

FIG. 2

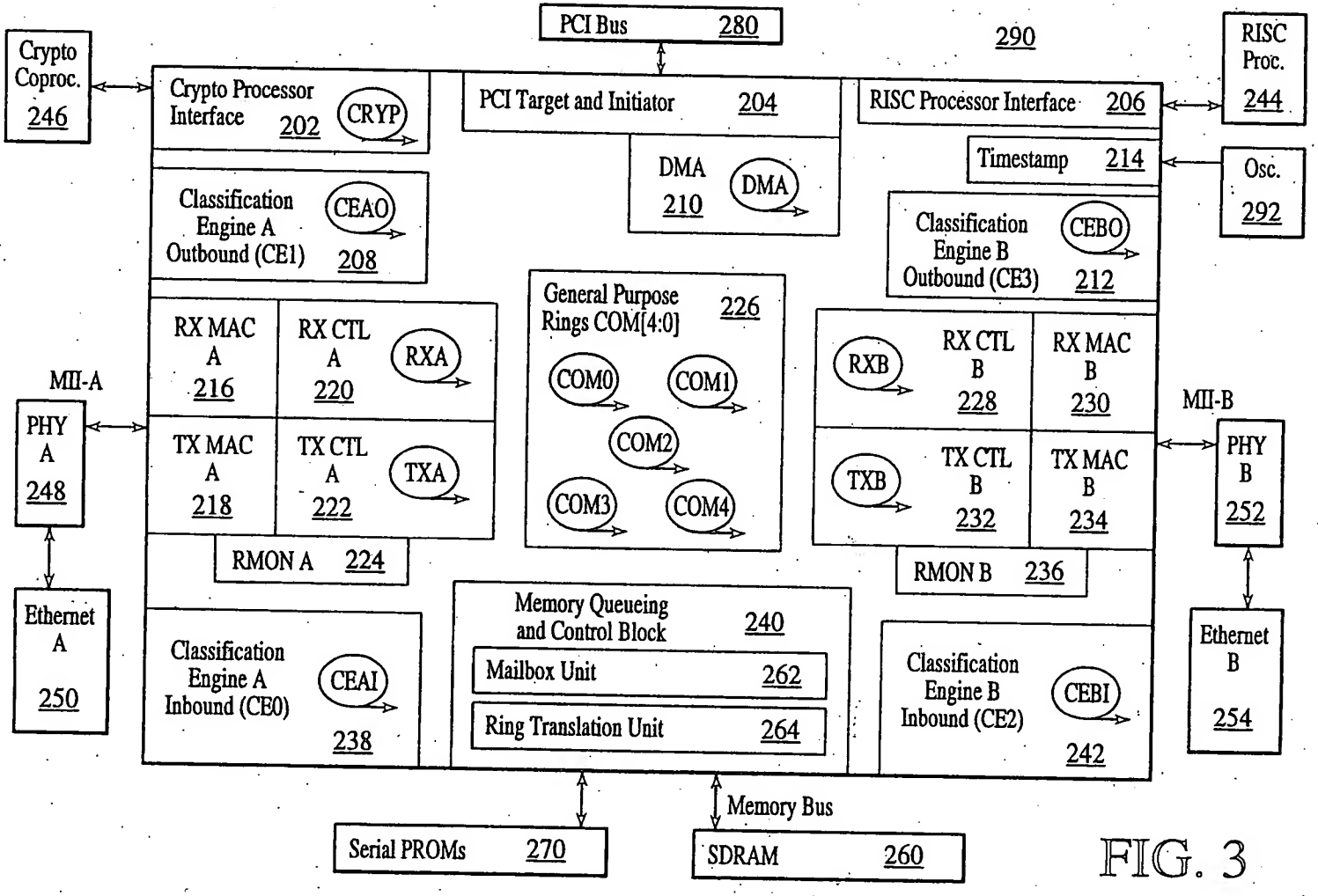

PCI Bus 280
290
Crypto Coproc. 246
Crypto Processor Interface 202
CRYP
PCI Target and Initiator 204
RISC Processor Interface 206
RISC Proc. 244
DMA 210
DMA
Timestamp 214
Osc. 292
Classification Engine A Outbound (CE1) 208
CEAO
Classification Engine B Outbound (CE3) 212
CEBO
General Purpose Rings COM[4:0] 226
COM0
COM1
COM2
COM3
COM4
RX MAC A 216
RX CTL A 220
RXA
TX MAC A 218
TX CTL A 222
TXA
RXB
RX CTL B 228
RX MAC B 230
TXB
TX CTL B 232
TX MAC B 234
MII-A
PHY A 248
MII-B
PHY B 252
RMON A 224
RMON B 236
Ethernet A 250
Ethernet B 254
Memory Queueing and Control Block 240
Mailbox Unit 262
Ring Translation Unit 264
Classification Engine A Inbound (CE0) 238
CEAI
Classification Engine B Inbound (CE2) 242
CEBI
Memory Bus
Serial PROMs 270
SDRAM 260

FIG. 3

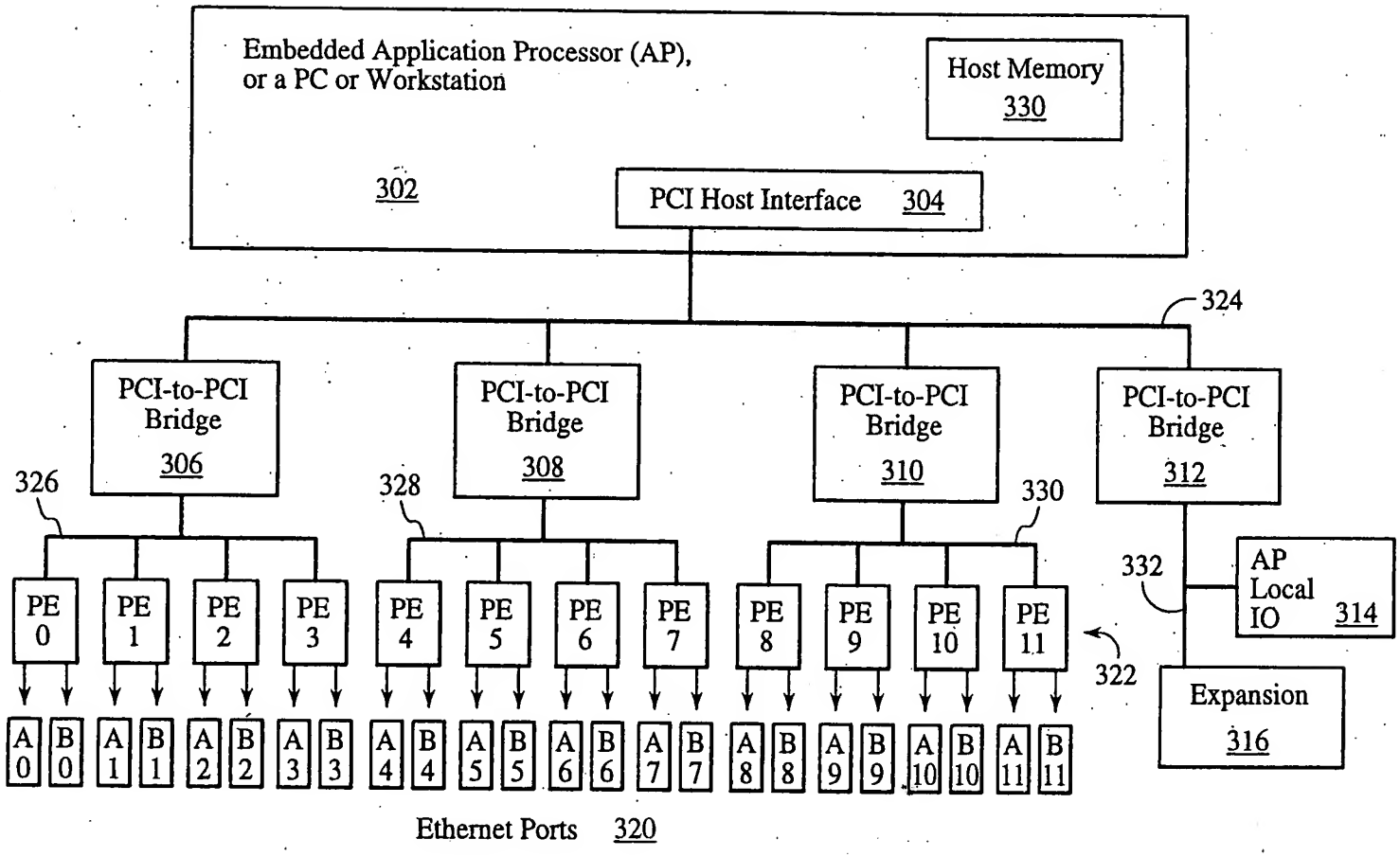

Embedded Application Processor (AP),
or a PC or Workstation
302
Host Memory
330
PCI Host Interface 304
324
PCI-to-PCI Bridge 306
PCI-to-PCI Bridge 308
PCI-to-PCI Bridge 310
PCI-to-PCI Bridge 312
326
328
330
332
AP Local IO 314
PE 0
PE 1
PE 2
PE 3
PE 4
PE 5
PE 6
PE 7
PE 8
PE 9
PE 10
PE 11
322
Expansion 316
A0 B0 A1 B1 A2 B2 A3 B3 A4 B4 A5 B5 A6 B6 A7 B7 A8 B8 A9 B9 A10 B10 A11 B11
Ethernet Ports 320

FIG. 4

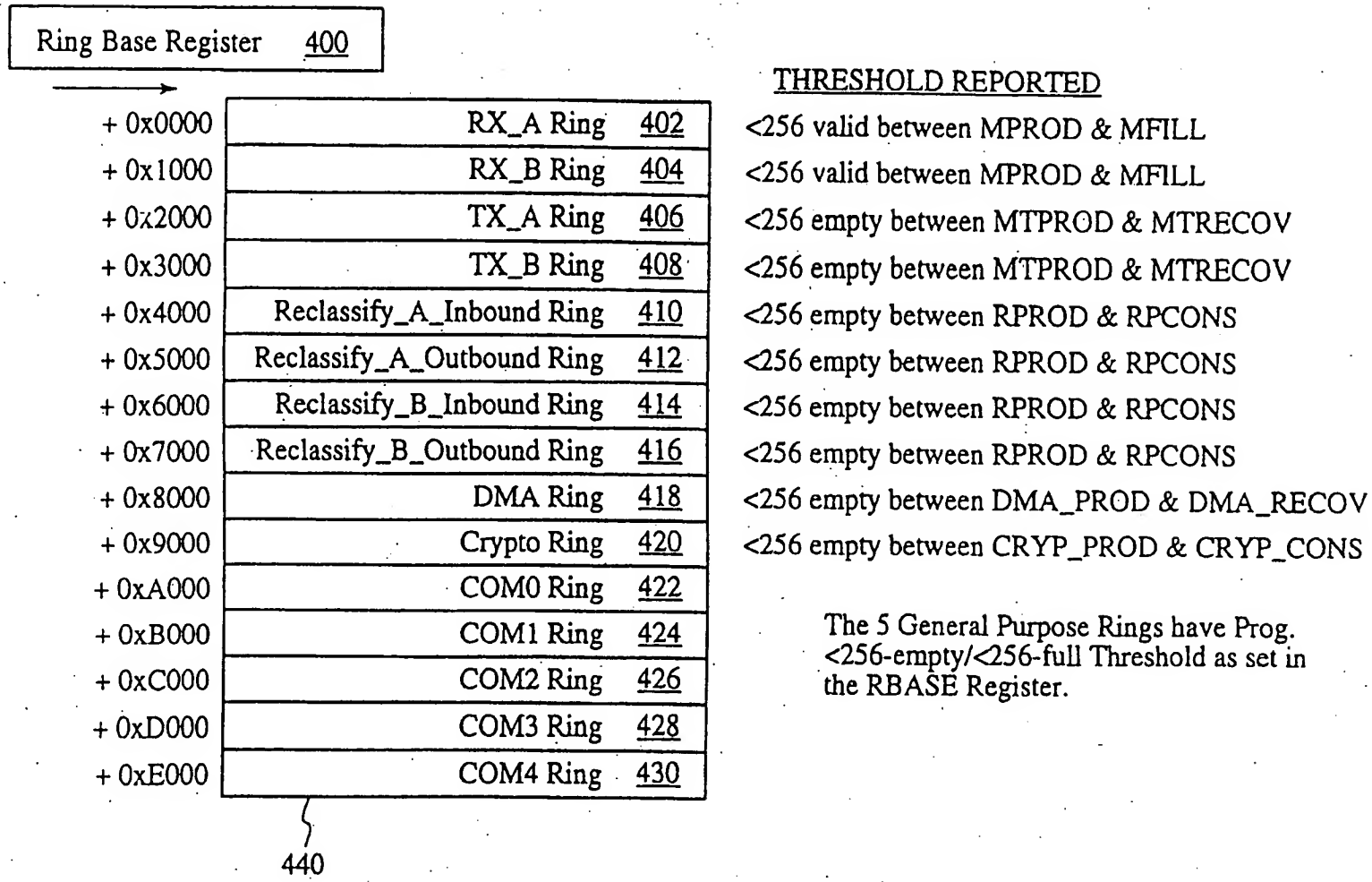

Ring Base Register 400
+ 0x0000
RX_A Ring 402
+ 0x1000
RX_B Ring 404
+ 0x2000
TX_A Ring 406
+ 0x3000
TX_B Ring 408
+ 0x4000
Reclassify_A_Inbound Ring 410
+ 0x5000
Reclassify_A_Outbound Ring 412
+ 0x6000
Reclassify_B_Inbound Ring 414
+ 0x7000
Reclassify_B_Outbound Ring 416
+ 0x8000
DMA Ring 418
+ 0x9000
Crypto Ring 420
+ 0xA000
COM0 Ring 422
+ 0xB000
COM1 Ring 424
+ 0xC000
COM2 Ring 426
+ 0xD000
COM3 Ring 428
+ 0xE000
COM4 Ring 430
440
THRESHOLD REPORTED
<256 valid between MPROD & MFILL
<256 valid between MPROD & MFILL
<256 empty between MTPROD & MTRECOV
<256 empty between MTPROD & MTRECOV
<256 empty between RPROD & RPCONS
<256 empty between RPROD & RPCONS
<256 empty between RPROD & RPCONS
<256 empty between RPROD & RPCONS
<256 empty between DMA_PROD & DMA_RECOV
<256 empty between CRYP_PROD & CRYP_CONS
The 5 General Purpose Rings have Prog. <256-empty/<256-full Threshold as set in the RBASE Register.

FIG. 5

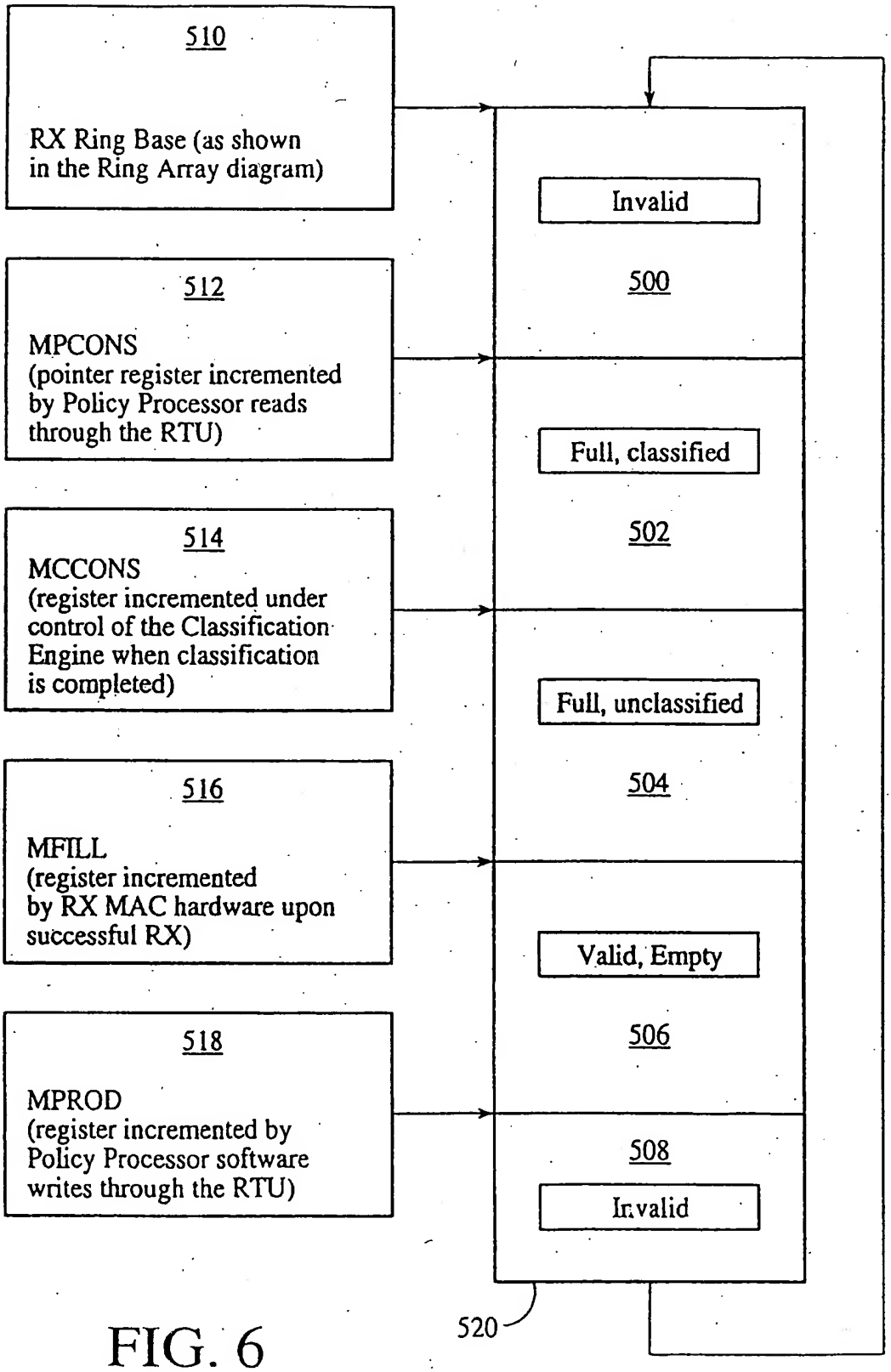

510
RX Ring Base (as shown in the Ring Array diagram)
512
MPCONS (pointer register incremented by Policy Processor reads through the RTU)
514
MCCONS (register incremented under control of the Classification Engine when classification is completed)
516
MFILL (register incremented by RX MAC hardware upon successful RX)
518
MPROD (register incremented by Policy Processor software writes through the RTU)
Invalid
500
Full, classified
502
Full, unclassified
504
Valid, Empty
506
508
Invalid
520

FIG. 6

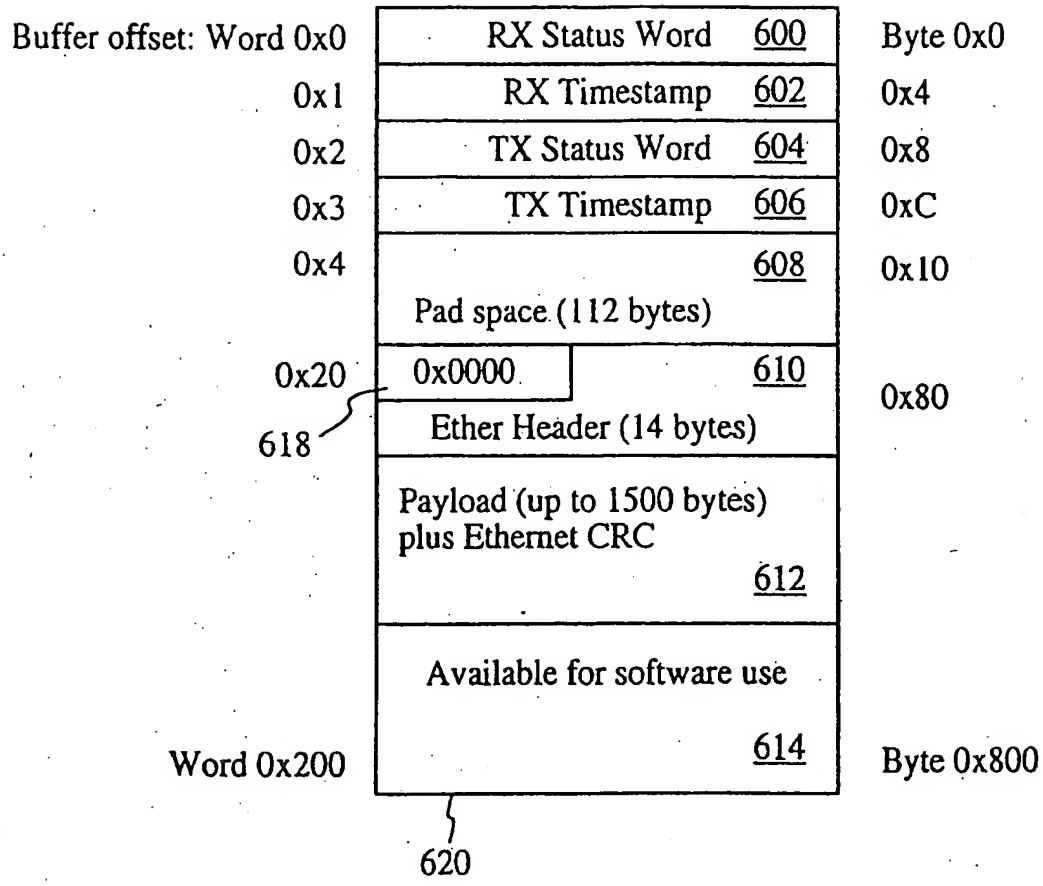

Buffer offset: Word 0x0
0x1
0x2
0x3
0x4
0x20
618
Word 0x200
620
RX Status Word 600
RX Timestamp 602
TX Status Word 604
TX Timestamp 606
608
Pad space (112 bytes)
0x0000
610
Ether Header (14 bytes)
Payload (up to 1500 bytes) plus Ethernet CRC
612
Available for software use
614
Byte 0x0
0x4
0x8
0xC
0x10
0x80
Byte 0x800

FIG. 7

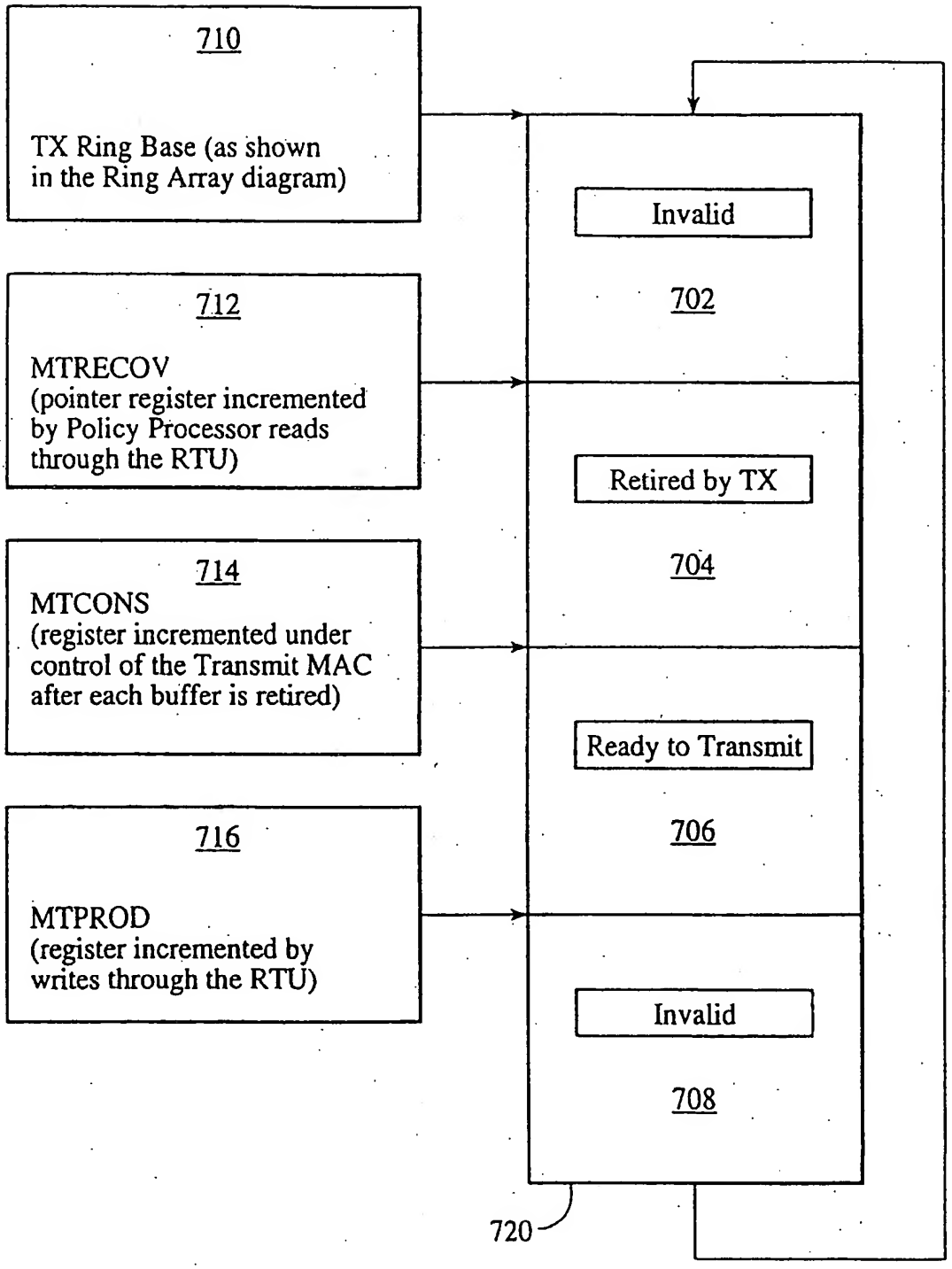

710
TX Ring Base (as shown in the Ring Array diagram)
712
MTRECOV (pointer register incremented by Policy Processor reads through the RTU)
714
MTCONS (register incremented under control of the Transmit MAC after each buffer is retired)
716
MTPROD (register incremented by writes through the RTU)
Invalid
702
Retired by TX
704
Ready to Transmit
706
Invalid
708
720

FIG. 8

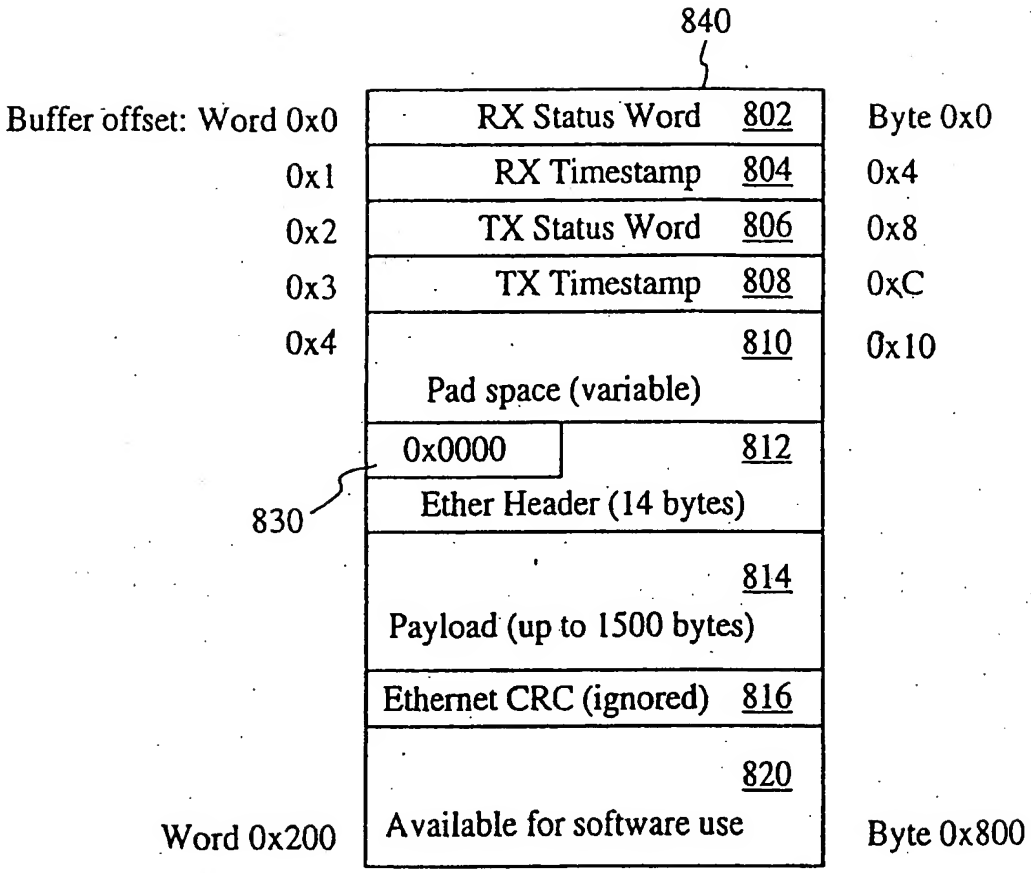

840
Buffer offset: Word 0x0
RX Status Word 802
Byte 0x0
0x1
RX Timestamp 804
0x4
0x2
TX Status Word 806
0x8
0x3
TX Timestamp 808
0xC
0x4
810
0x10
Pad space (variable)
0x0000
812
830
Ether Header (14 bytes)
814
Payload (up to 1500 bytes)
Ethernet CRC (ignored) 816
820
Word 0x200
Available for software use
Byte 0x800

FIG. 9

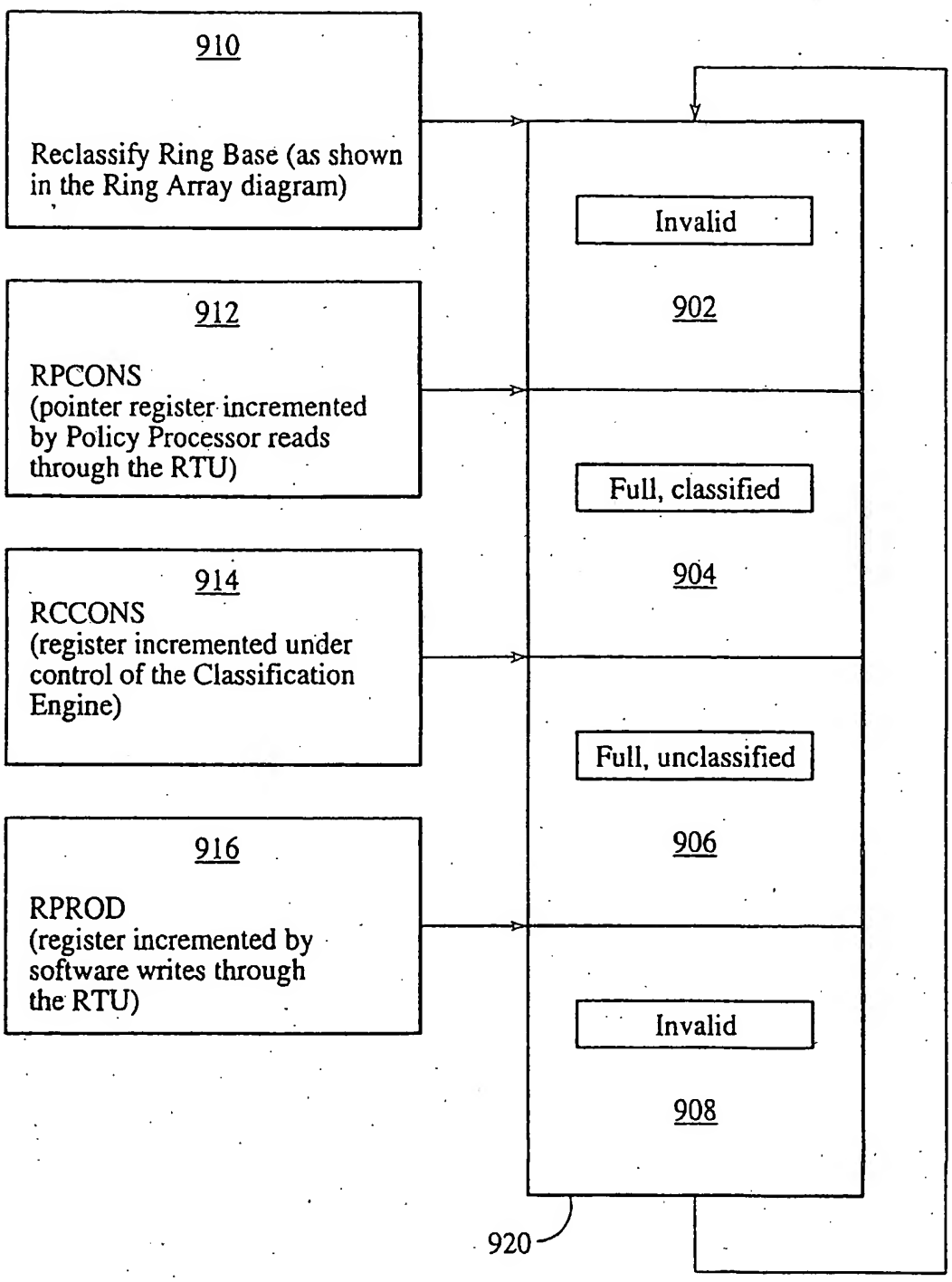

910
Reclassify Ring Base (as shown in the Ring Array diagram)
912
RPCONS
(pointer register incremented by Policy Processor reads through the RTU)
914
RCCONS
(register incremented under control of the Classification Engine)
916
RPROD
(register incremented by software writes through the RTU)
Invalid
902
Full, classified
904
Full, unclassified
906
Invalid
908
920

FIG. 10

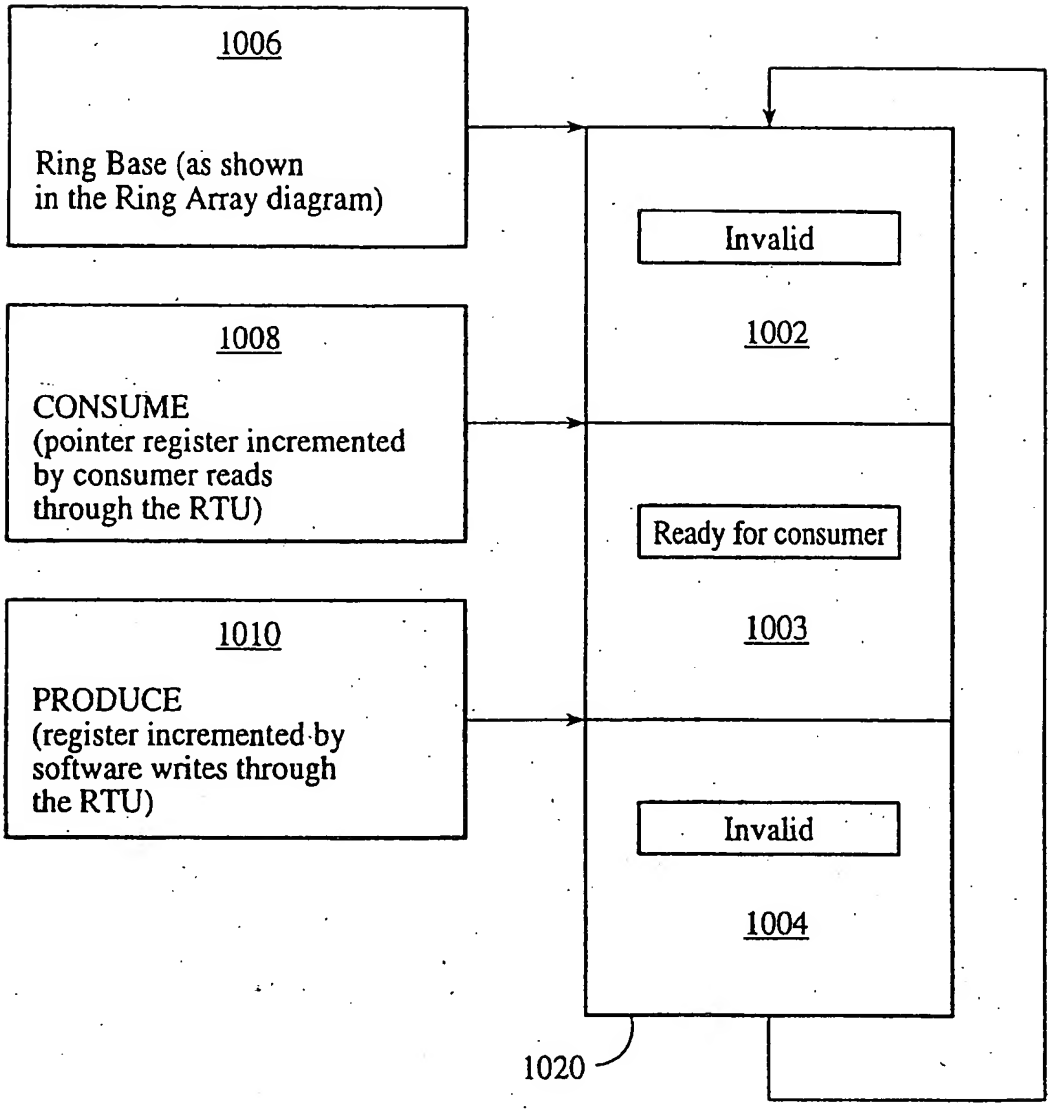

1006
Ring Base (as shown in the Ring Array diagram)
1008
CONSUME (pointer register incremented by consumer reads through the RTU)
1010
PRODUCE (register incremented by software writes through the RTU)
Invalid
1002
Ready for consumer
1003
Invalid
1004
1020

FIG. 11

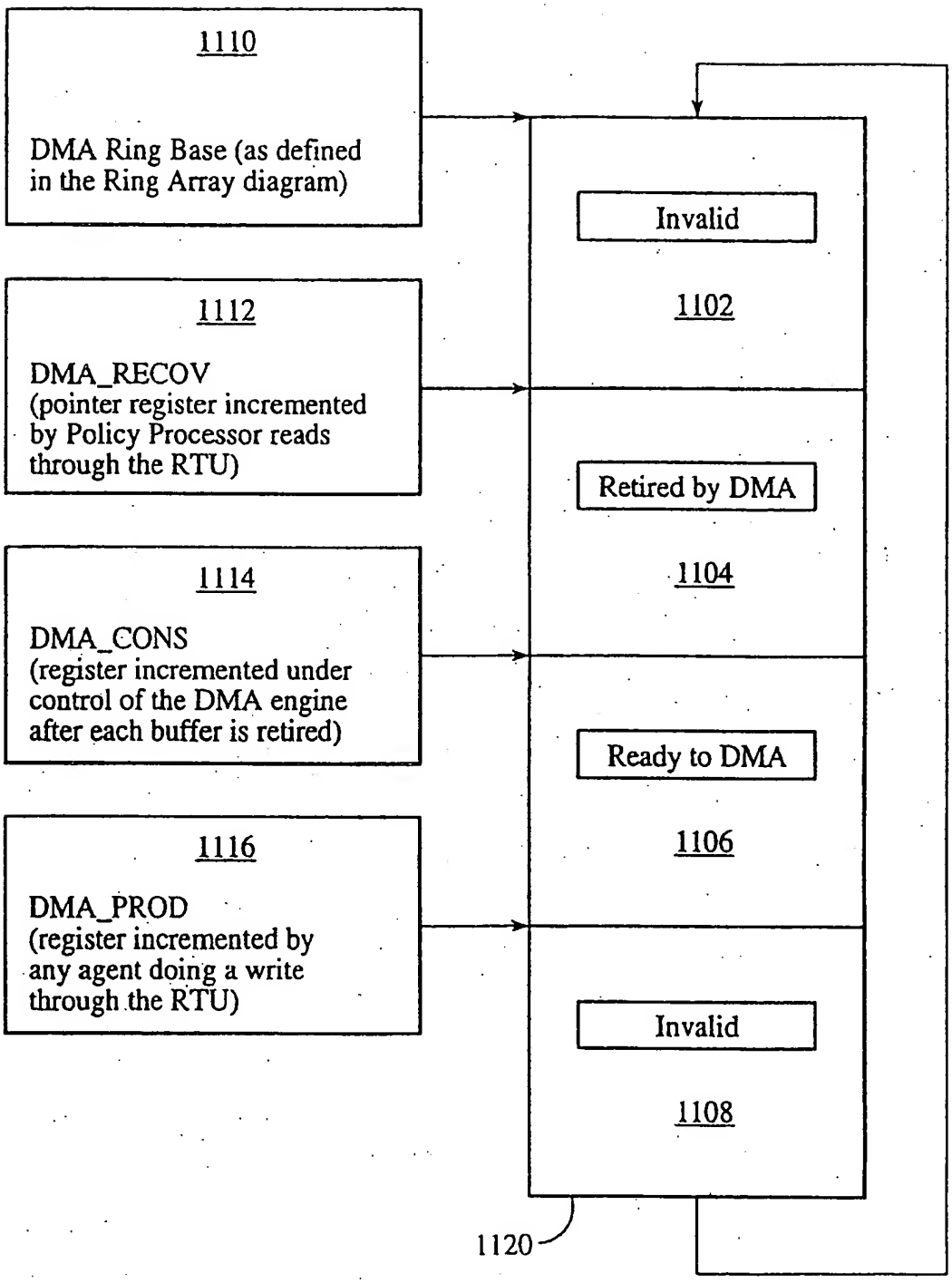
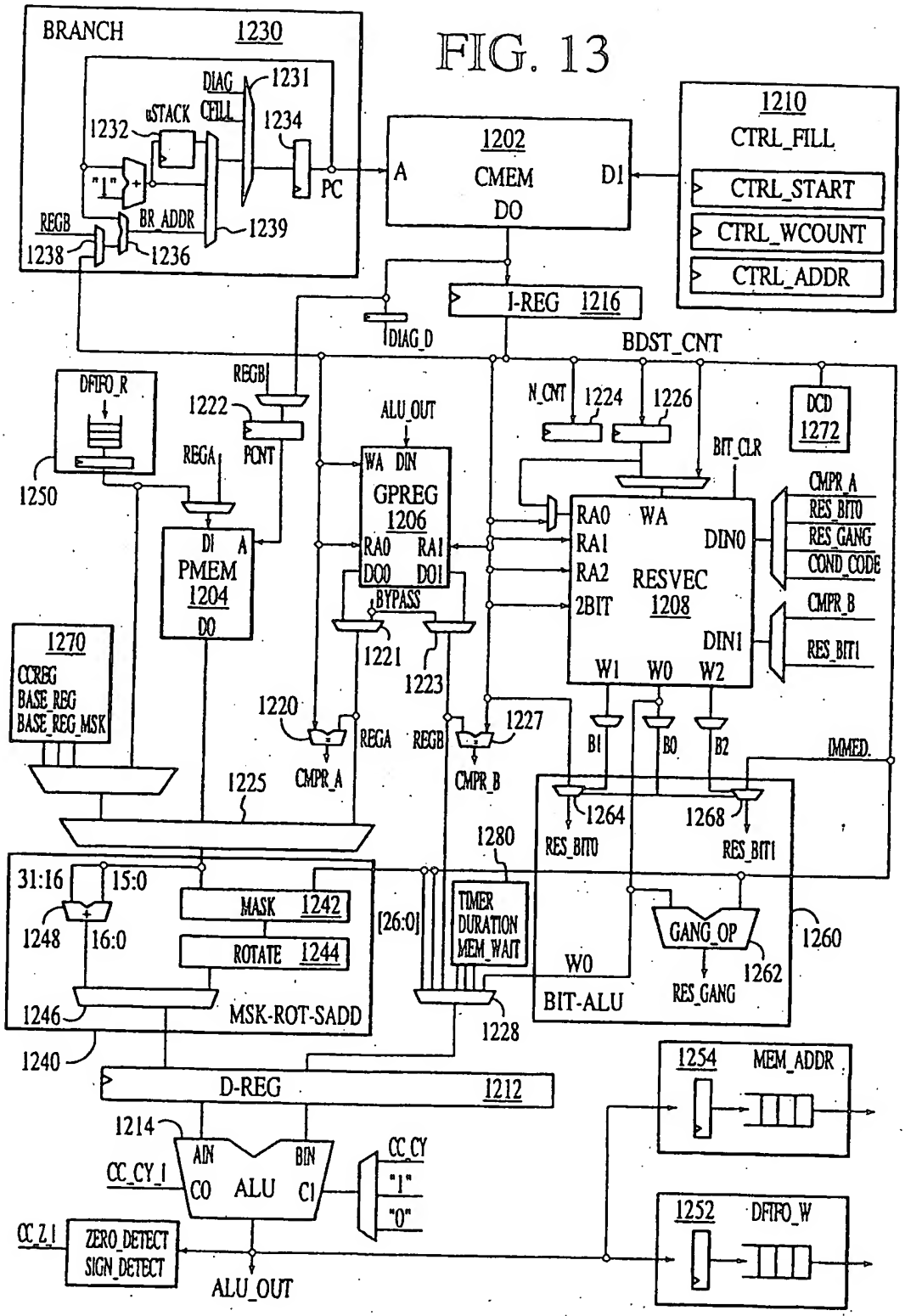

1110
DMA Ring Base (as defined in the Ring Array diagram)
1112
DMA_RECOV (pointer register incremented by Policy Processor reads through the RTU)
1114
DMA_CONS (register incremented under control of the DMA engine after each buffer is retired)
1116
DMA_PROD (register incremented by any agent doing a write through the RTU)
Invalid
1102
Retired by DMA
1104
Ready to DMA
1106
Invalid
1108
1120

FIG. 12

FIG. 13


BRANCH 1230
DIAG 1231
uSTACK CELL
1232
1234
"1"
PC
REGB
BR_ADDR
1239
1238
1236
1202
CMEM
A
DO
DI
1210
CTRL_FILL
CTRL_START
CTRL_WCOUNT
CTRL_ADDR
I-REG 1216
DIAG_D
BDST_CNT
DFIFO_R
1250
REGB
1222
PCNT
REGA
ALU_OUT
WA DIN
GPREG
1206
RA0 RA1
DO0 DO1
N_CNT 1224 1226
DCD
1272
BIT_CLR
CMPR_A
RES_BIT0
RES_GANG
COND_CODE
CMPR_B
RES_BIT1
RA0 WA
RA1
RA2
2BIT
RESVEC
1208
DIN0
DIN1
W1 W0 W2
DI A
PMEM
1204
DO
BYPASS
1221
1223
1270
CCREG
BASE_REG
BASE_REG_MSK
1220
REGA
REGB
1227
CMPR_A
CMPR_B
B1
B0
B2
IMMED.
1225
1264
1268
RES_BIT0
RES_BIT1
1280
31:16
15:0
1248
16:0
MASK 1242
ROTATE 1244
[26:0]
TIMER
DURATION
MEM_WAIT
GANG_OP
1260
1262
RES_GANG
W0
BIT-ALU
1246
MSK-ROT-SADD
1228
1240
D-REG 1212
1254
MEM_ADDR
1214
AIN
BIN
CC_CY
CC_CY_I
CO
ALU
CI
"1"
"0"
1252
DFIFO_W
CC_Z_I
ZERO_DETECT
SIGN_DETECT
ALU_OUT

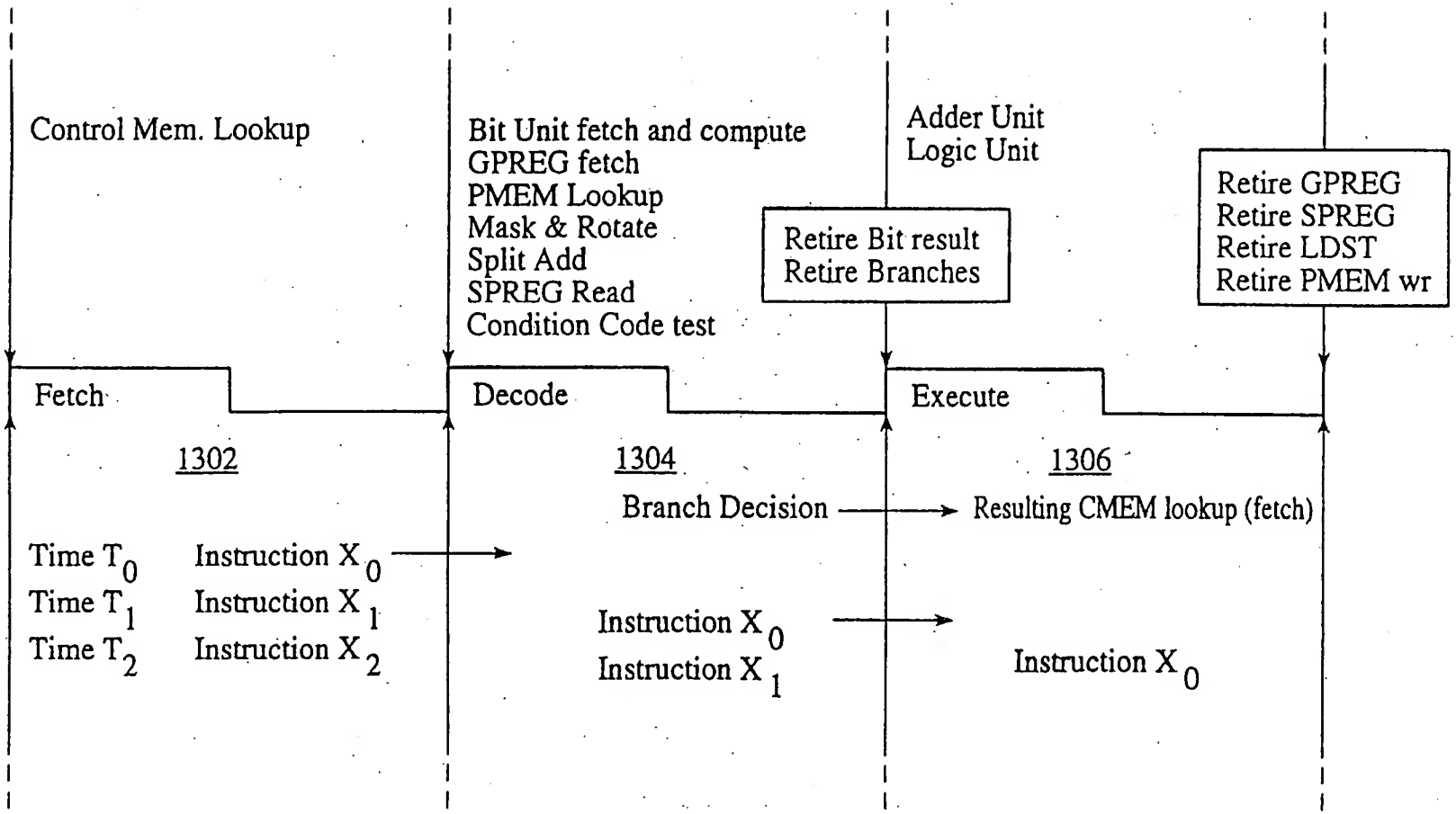

Control Mem. Lookup
Bit Unit fetch and compute
GPREG fetch
PMEM Lookup
Mask & Rotate
Split Add
SPREG Read
Condition Code test
Adder Unit
Logic Unit
Retire Bit result
Retire Branches
Retire GPREG
Retire SPREG
Retire LDST
Retire PMEM wr
Fetch
Decode
Execute
1302
1304
1306
Branch Decision
Resulting CMEM lookup (fetch)
Time T0
Time T1
Time T2
Instruction X0
Instruction X1
Instruction X2
Instruction X0
Instruction X1
Instruction X0

FIG. 14

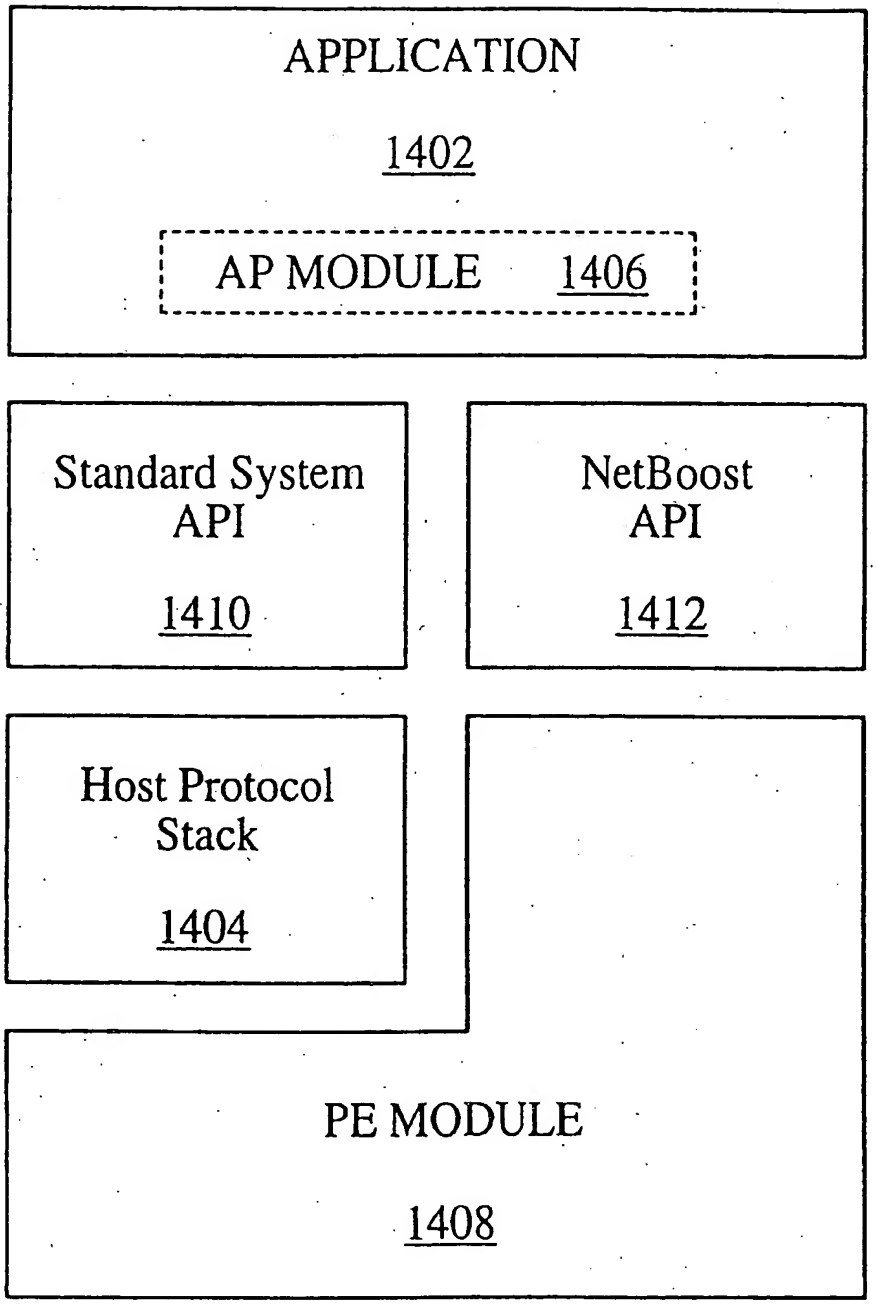

APPLICATION
1402
AP MODULE 1406
Standard System API
1410
NetBoost API
1412
Host Protocol Stack
1404
PE MODULE
1408

FIG. 15

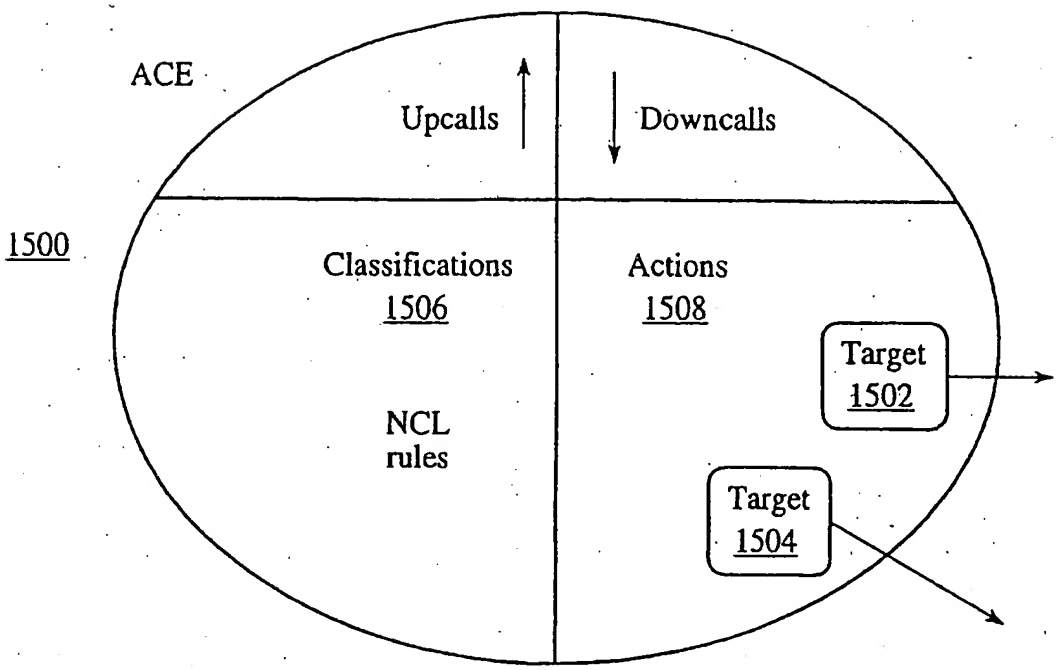

ACE
Upcalls
Downcalls
1500
Classifications
1506
Actions
1508
Target
1502
NCL
rules
Target
1504

FIG. 16

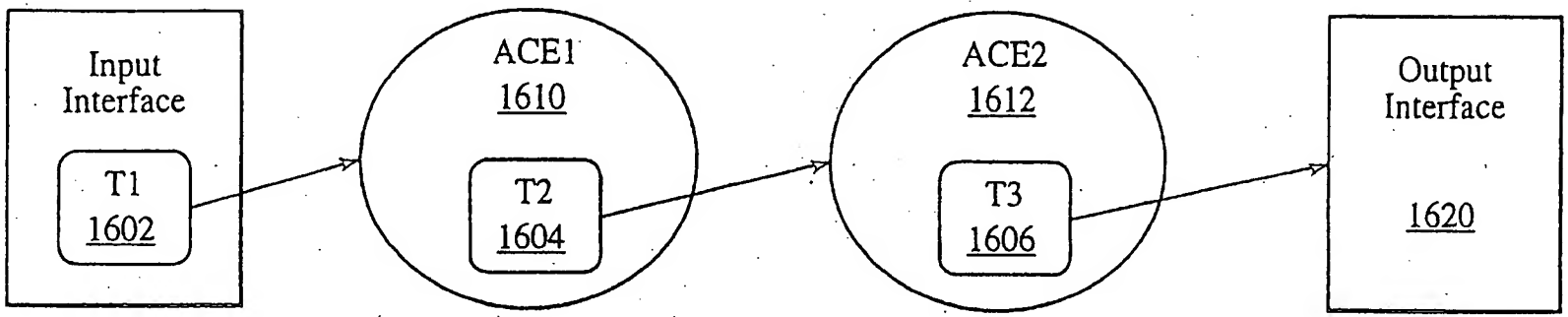

Input
Interface
T1
1602
ACE1
1610
T2
1604
ACE2
1612
T3
1606
Output
Interface
1620

FIG. 17

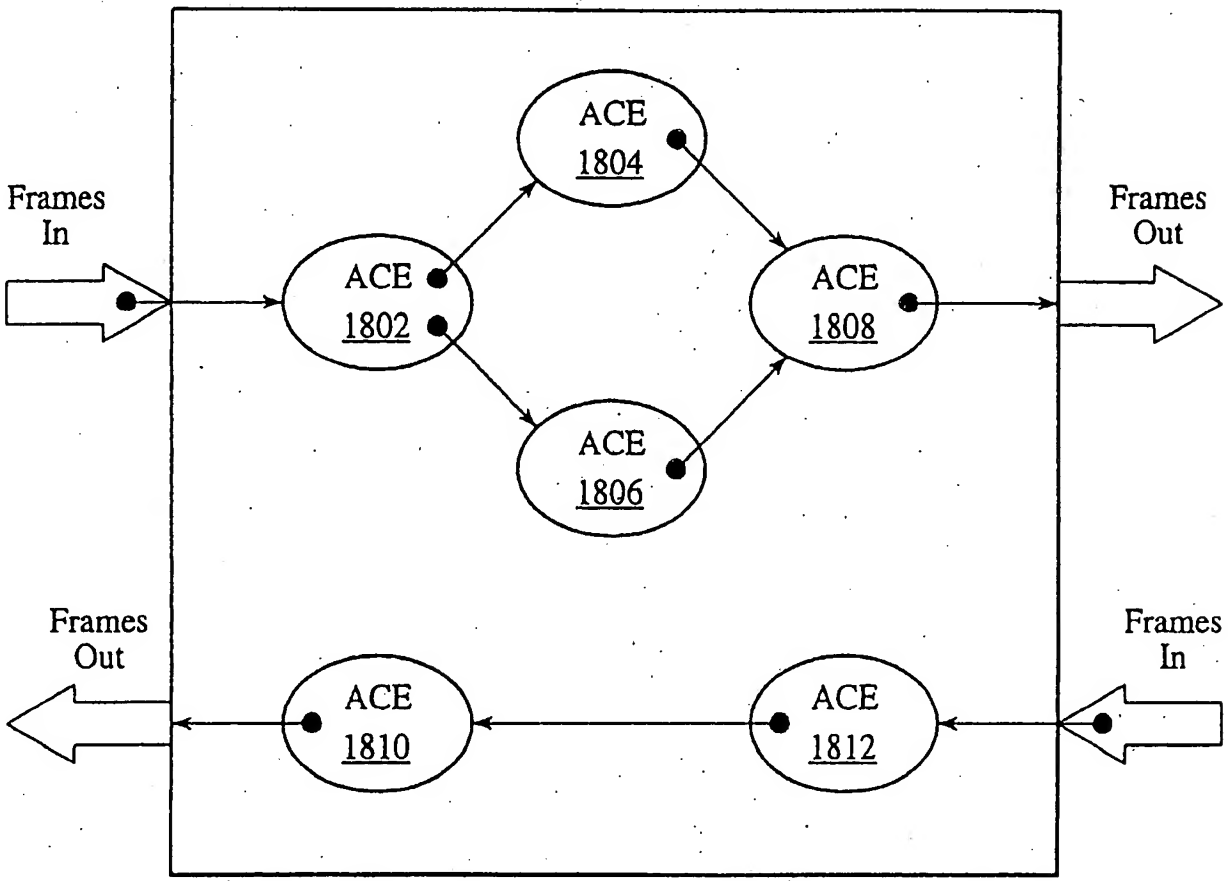

Frames
In
ACE
1804
ACE
1802
ACE
1808
ACE
1806
Frames
Out
Frames
Out
ACE
1810
ACE
1812
Frames
In

FIG. 18

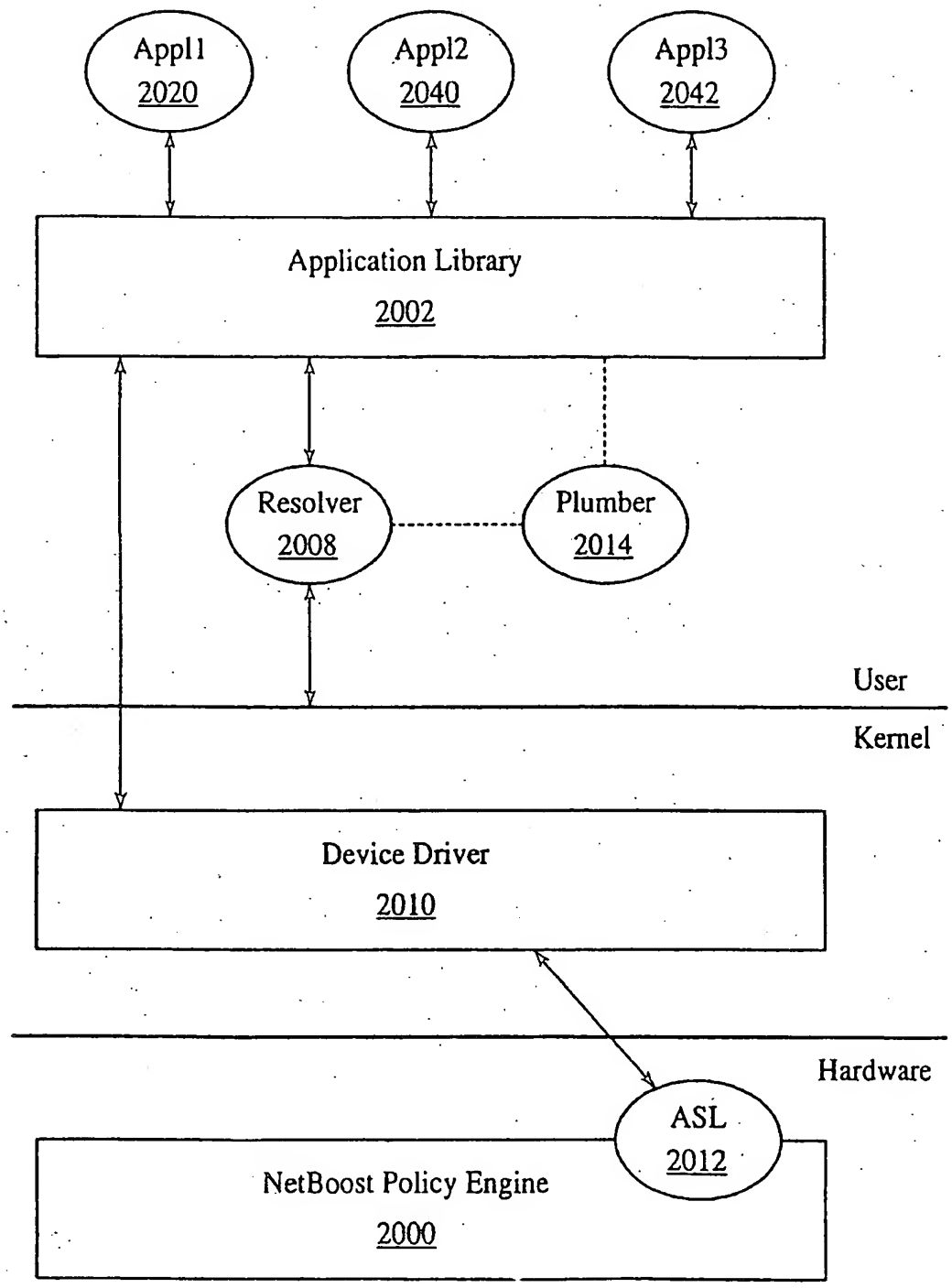

Appl1
2020
Appl2
2040
Appl3
2042
Application Library
2002
Resolver
2008
Plumber
2014
User
Kernel
Device Driver
2010
Hardware
ASL
2012
NetBoost Policy Engine
2000

FIG. 19